

JAVA.LANG.CLASSLOADER CLASS

http://www.tutorialspoint.com/java/lang/java_lang_classloader.htm

Copyright © tutorialspoint.com

Introduction

The **java.lang.ClassLoader** class is an object that is responsible for loading classes. This class is an abstract class. It may be used by security managers to indicate security domains.

Class declaration

Following is the declaration for **java.lang.ClassLoader** class:

```
public abstract class ClassLoader
    extends Object
```

Class constructors

S.N.	Constructor & Description
1	protected ClassLoader() This creates a new class loader using the ClassLoader returned by the method <code>getSystemClassLoader()</code> as the parent class loader.
2	protected ClassLoader(ClassLoader parent) This creates a new class loader using the specified parent class loader for delegation.

Class methods

S.N.	Method & Description
1	<u>void clearAssertionStatus()</u> This method sets the default assertion status for this class loader to false and discards any package defaults or class assertion status settings associated with the class loader.
2	<u>protected Class<?> defineClass(String name, byte[] b, int off, int len)</u> This method converts an array of bytes into an instance of class Class.
3	<u>protected Class<?> defineClass(String name, byte[] b, int off, int len, ProtectionDomain protectionDomain)</u> This method converts an array of bytes into an instance of class Class, with an optional ProtectionDomain
4	<u>protected Class<?> defineClass(String name, ByteBuffer b, ProtectionDomain protectionDomain)</u> This method converts a ByteBuffer into an instance of class Class, with an optional ProtectionDomain.
5	<u>protected Package definePackage(String name, String specTitle, String specVersion, String specVendor, String implTitle, String implVersion, String implVendor, URL sealBase)</u> This method defines a package by name in this ClassLoader.
6	<u>protected Class<?> findClass(String name)</u> This method finds the class with the specified binary name.

7	<u>protected String findLibrary(String libname)</u> This method returns the absolute path name of a native library.
8	<u>protected Class<?> findLoadedClass(String name)</u> This method returns the class with the given binary name if this loader has been recorded by the Java virtual machine as an initiating loader of a class with that binary name.
9	<u>protected URL findResource(String name)</u> This method Finds the resource with the given name.
10	<u>protected Enumeration<URL> findResources(String name)</u> This method returns an enumeration of URL objects representing all the resources with the given name.
11	<u>protected Class<?> findSystemClass(String name)</u> This method finds a class with the specified binary name, loading it if necessary.
12	<u>protected Package getPackage(String name)</u> This method returns a Package that has been defined by this class loader or any of its ancestors.
13	<u>protected Package[] getPackages()</u> This method returns all of the Packages defined by this class loader and its ancestors.
14	<u>ClassLoader getParent()</u> This method returns the parent class loader for delegation.
15	<u>URL getResource(String name)</u> This method finds the resource with the given name.
16	<u>InputStream getResourceAsStream(String name)</u> This method returns an input stream for reading the specified resource.
17	<u>Enumeration<URL> getResources(String name)</u> This method finds all the resources with the given name.
18	<u>static ClassLoader getSystemClassLoader()</u> This method returns the system class loader for delegation.
19	<u>static URL getSystemResource(String name)</u> This method find a resource of the specified name from the search path used to load classes.
20	<u>static InputStream getSystemResourceAsStream(String name)</u> This method is open for reading, a resource of the specified name from the search path used to load classes.
21	<u>static Enumeration<URL> getSystemResources(String name)</u> This method finds all resources of the specified name from the search path used to load classes.
22	<u>Class<?> loadClass(String name)</u> This method loads the class with the specified binary name.
23	<u>protected Class<?> loadClass(String name, boolean resolve)</u> This method loads the class with the specified binary name.
24	<u>protected void resolveClass(Class<?> c)</u> This method links the specified class.
25	<u>void setClassAssertionStatus(String className, boolean enabled)</u> This method sets the desired assertion status for the named top-level class in this class loader and any nested

	classes contained therein.
26	<u>void setDefaultAssertionStatus(boolean enabled)</u> This method sets the default assertion status for this class loader.
27	<u>void setPackageAssertionStatus(String packageName, boolean enabled)</u> This method sets the package default assertion status for the named package.
28	<u>protected void setSigners(Class<?> c, Object[] signers)</u> This method sets the signers of a class.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object